On L_1-norm Multi-class Support Vector Machines [*]

Lifeng Wang [†] Xiaotong Shen [‡] Yuan Zheng [§]

Abstract

Binary Support Vector Machines (SVM) have proven effective in classification. However, problems remain with respect to feature selection in multi-class classification. This article proposes a novel multi-class SVM, which performs classification and feature selection simultaneously via L_1-norm penalized sparse representations. The proposed methodology, together with our developed regularization solution path, permits feature selection within the framework of classification. The operational characteristics of the proposed methodology is examined via both simulated and benchmark examples, and is compared to some competitors in terms of the accuracy of prediction and feature selection. The numerical results suggest that the proposed methodology is highly competitive.

1 Introduction

Binary support vector machines (SVM, Boser, Guyon and Vapnik, 1992; Cortes and Vapnik, 1995), as a powerful classification tool, have proven effective in achieving the state-of-the-art performance in various applications. For feature selection in classification, Bradley and Mangasarian (1998) introduced a SVM with a L_1-norm penalty. Challenges remain with regard to multi-class classification, particularly for high-dimensional problems as well as for understanding feature selection within the framework of classification. In this article, we develop a novel L_1-norm multi-class SVM and investigate its feasibility in feature selection.

In multi-class classification, a common treatment is "one-versus-all" approach (OVA), which trains a sequence of binary classifiers to distinguish one class from the remaining classes. Based on OVA, Shawe-Taylor, Saunders and Hardoon (2004) generalizes the result of Bradley and Mangasarian (1998). While OVA delivers good empirical performances, it has three potential drawbacks. First, with respect to training, OVA trains binary decision classifiers sequentially. When the number of classes becomes large, each binary classification becomes highly unbalanced, with a small fraction of instances in one class. In the case of nonseparable classes, the class with smaller fraction of instances tends to be ignored, leading to degraded performances. Second, with respect to feature selection in OVA, one feature is relevant for all classes if it is selected in one binary classification. For a sparse problem in presence of many irrelevant features, this usually results in more than necessary features and has an adverse effect on classification, because these redundant features have not eliminated. Further investigation is needed to understand feature selection in classification. Third, with respect to fisher consistency, it is unclear if OVA targets the Bayes rule for a given class of candidate decision functions. Although it is well known that the binary SVM estimates Bayes rule, the combined classifier in multi-class case may not target Bayes rule in absence of dominating class, in view of the results of Lee, Lin and Wahba (2004).

This article proposes an L_1-norm MSVM (L1MSVM), which attempts to attack feature selection by treating multiple classes jointly, as opposed to "one-versus-all", in multi-class classification. L1MSVM overcomes the aforementioned difficulties of OVA, in addition to generalizing the concepts of margin. It has the ability of performing feature selection and classification simultaneously, while retaining the margin interpretability of its L_2-norm counterpart. Moreover, dimension reduction is built into classification, bypassing the requirement of an ad hoc step of dimension reduction to attack a large problem that is beyond the capability of conventional techniques. This is the case in cancer genomic classification, where gene pre-screening is required, e.g., Dudoit, Fridlyand and Speed (2002), and Guyon and Elisseeff (2003).

Key to the performance of L1MSVM is data-adaptive tuning. In practice, tuning usually requires solving L1MSVM repeatedly at each value of tuning parameter, which is computationally expensive, particularly for high-dimensional problems. To reduce the computational cost, we develop an efficient algorithm to solve an entire solution path as a function of the tuning parameter with a complexity of the same order as solving a single L1MSVM, which facilitates adaptive

[*]Research is supported in part by NSF Grant IIS-0328802.

[†]School of Statistics, The University of Minnesota, Minneapolis MN 55455. Email: iamwlf@stat.umn.edu

[‡]School of Statistics, The University of Minnesota, Minneapolis MN 55455. Email: xshen@stat.umn.edu

[§]Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210.

selection of the tuning parameter.

L1MSVM, together with our developed regularization path, leads to efficient computation for large problems. Our numerical result indicates that L1MSVM is highly competitive against OVA.

This article is organized as follows. Section 2 briefly introduces the methodology and the algorithm. Some numerical results on both simulated and real examples are presented in Section 3, followed by a summary in Section 4.

2 Methodology

In classification, a training sample $z_i = (x_i, y_i)$; $i = 1, \ldots, n$ is given, which is sampled from an unknown distribution $P(x, y)$, with input $x_i \in \mathrm{R}^p$ a vector of p predictors, and output y_i indicating the class label. In k-class classification, y is usually coded as $\{1, 2, \ldots, k\}$, in addition to a decision function vector $f = (f_1, \ldots, f_k)$, with f_c representing class c; $c = 1, \ldots, k$. To avoid redundancy in f, a zero-sum constraint $\sum_{c=1}^k f_c = 0$ is enforced. Given f, a classification rule is $\Phi_f(x) = \arg\max_c f_c(x)$, which assigns a new input vector x to class c having the highest value $f_c(x)$. Our goal is to seek f that minimizes the generalization error (GE), defined as $\mathrm{Err}(f) = E(I[Y \neq \Phi_f(x)])$, based on $z_i = (x_i, y_i)$; $i = 1 \ldots n$.

For motivation, we first discuss the L_1-norm binary SVM (L1SVM) with $k = 2$ and $y \in \{-1, +1\}$.

2.1 Motivation: binary classification. For motivation, we begin our discussion with the binary L_1-norm SVM (L1SVM) with $Y \in \{-1, +1\}$. In this case, SVM uses an p-dimensional hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ as a decision function, with the corresponding decision rule $\Phi(\boldsymbol{x}) = \mathrm{sign}(f(\boldsymbol{x}))$. Bradley et al. (1998) proposed L1SVM in the form of

$$\min_{w,b} V(y_i f(\boldsymbol{x}_i)) + \lambda \|\boldsymbol{w}\|_1, \tag{2.1}$$

where $V(z) = [1 - z]_+$ is the hinge loss (c.f., Wahba 1999), and $\|\boldsymbol{w}\|_1 = \sum_{j=1}^p |w_j|$ is the L_1-norm of $\boldsymbol{w}$. In the linear separable case, (2.1) can be thought of as maximizing the geometric margin $\frac{2}{\|\boldsymbol{w}\|_1}$, which is the L_∞-distance between two hyperplanes $\boldsymbol{w}^T\boldsymbol{x} + b = \pm 1$, defined as $\inf_{\boldsymbol{x},\boldsymbol{x}'}\{\|\boldsymbol{x} - \boldsymbol{x}'\|_\infty : \boldsymbol{w}^T\boldsymbol{x} + b = 1, \boldsymbol{w}^T\boldsymbol{x}' + b = -1\}$ with $\|\boldsymbol{x}\|_\infty = \max_{1\le j\le p} |x_j|$ the L_∞-norm. Figure 1 illustrates the subtle difference between L1SVM and SVM with respect to the choice of metric in the linear separable case. In contrast to the standard SVM having a representation with sparse support vectors, L1SVM have sparse features in that the number of non-zero coefficients of w is small, which enables L1SVM to perform feature selection and classification simultaneously.

To extend (2.1) to the multi-class case, we need to generalize the hinge loss as well as the L_1-penalty in the binary case. In the literature, several generalizations of the binary hinge loss exist in the different context. Vapnik (1998), Weston and Watkins (1998), Bredensteiner and Bennett (1999), and Guermuer (2002) used a generalized hinge loss in the form of

$$V(f, z_i) = \sum_{c \neq y_i} [1 - (f_{y_i}(x_i) - f_c(x_i))]_+. \tag{2.2}$$

Liu and Shen (2005) suggested

$$V(f, z_i) = [1 - \min_c (f_{y_i}(x_i) - f_c(x_i))]_+. \tag{2.3}$$

Lee et al. (2004) proposed

$$V(f, z_i) = \sum_{c \neq y_i} [f_c(x_i) + 1]_+, \tag{2.4}$$

which seems to have a global Fisher consistency property for MSVM with the L_2-norm. However, it remains unclear if it leads to sharp generalization in our case.

In this article, we propose a novel L_1-norm MSVM via (2.4), although our framework is readily applicable to other formulations straightforwardly.

2.2 L1MSVM. We use linear decision functions $f_c(x) = w_c^T x + b_c$; $c = 1, \ldots, k$ are linear, with $w_c = (w_{c,1}, \ldots, w_{c,p}) \in \mathrm{R}^p$ and $b_c \in \mathrm{R}^1$ subject to zero-sum constraints $\sum_{c=1}^k w_c = \vec{0}$ and $\sum_{c=1}^k b_c = 0$. In nonlinear classification, decision functions $f_c(x) = \sum_{j=1}^q w_{c,j} h_j(x) + b_c$; $c = 1, \ldots, k$ have flexible representations on a basis $\{h_j(x)\}_{j=1}^q$, with $H = (h_j(x_i))_{n\times q}$ being a design matrix. For the purpose of feature selection, we use linear representations and proposed L_1-norm MSVM (L1MSVM):

$$\begin{aligned} &\min_{w_c, b_c; c=1,\ldots,k} && \sum_{i=1}^n V(f, z_i), \\ &\text{subject to} && \sum_{c=1}^k \|w_c\|_1 \le s \\ &\text{and} && \sum_c w_c = \vec{0}; \sum_c b_c = 0, \end{aligned} \tag{2.5}$$

where $\sum_{c=1}^k \|w_c\|_1 = \sum_{c=1}^k \sum_{j=1}^p |w_{c,j}|$ is an L_1-norm penalty, s is a tuning parameter, and $V(f, z_i)$ is the generalized hinge loss in (2.4).

For any given value of s, (2.5) can be treated via linear programming by solving

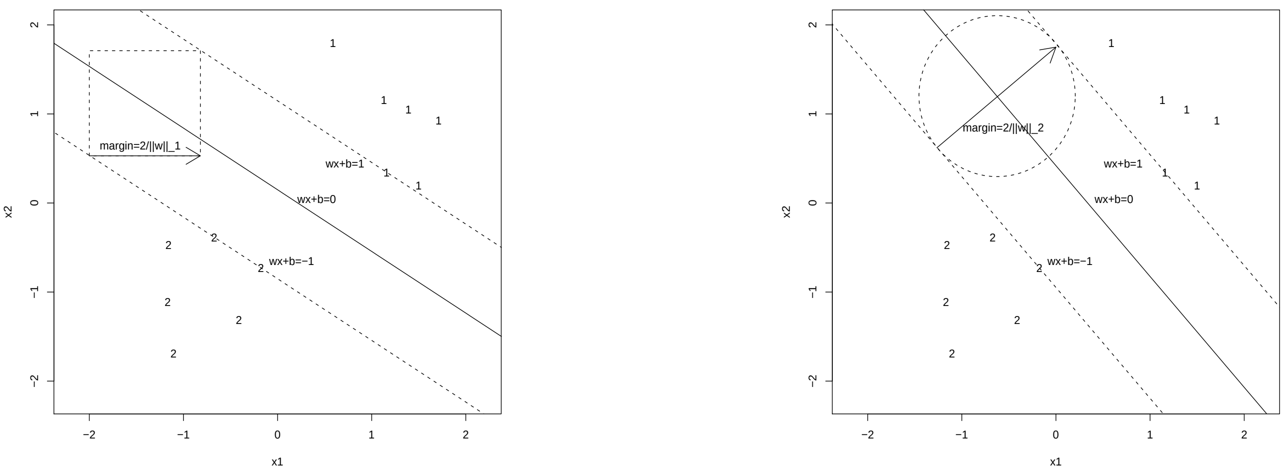

Figure 1: Illustration of L1SVM and SVM in the linear separable case. Left: L1SVM maximizes the L_∞-margin; Right: SVM maximizes the L_2-margin.

$$
\begin{aligned}
&\min_{w^+_{c,j}, w^-_{c,j}, b^+_c, b^-_c, \xi_l} \sum_{l=1}^{n(k-1)} \xi_l \\
&\text{subject to} \\
&\sum_j (w^+_{c,j} - w^-_{c,j}) x_{ij} + (b^+_c - b^-_c) + 1 \le \xi_l; \\
&\sum_{c,j} (w^+_{c,j} + w^-_{c,j}) \le s, \\
&\sum_c (w^+_{c,j} - w^-_{c,j}) = 0; \forall j, \\
&\sum_c (b^+_c - b^-_c) = 0. \\
&w^+_{c,j}, w^-_{c,j}, b^+_c, b^-_c, \xi_l \ge 0,
\end{aligned}
\tag{2.6}
$$

which can be computed by a standard package such as "lpsolve" in R. The solution of (2.5) $\hat{w}_{c,j}$ and $\hat{b}_c$ can be obtained by $\hat{w}_{c,j} = \hat{w}^+_{c,j} - \hat{w}^-_{c,j}$ and $\hat{b}_c = \hat{b}^+_c - \hat{b}^-_c$; $c = 1, \ldots, k$, $j = 1, \ldots, p$, where $\hat{w}^+_{c,j}, \hat{w}^-_{c,j}, \hat{b}^+_c$,and $\hat{b}^-_c$ are the solutions of (2.6). This yields $\hat{f}(x) = (\hat{w}_1^T x + \hat{b}_1, \ldots, \hat{w}_k^T x + \hat{b}_k)$ and the corresponding $\Phi(x) = \arg\max_c(\hat{w}_c x + \hat{b}_c)$.

L1MSVM can be cast into the framework of regularization as follows:

$$
\begin{aligned}
\min_{w_c, b_c; c=1,\ldots,k} \quad & \sum_{i=1}^n V(f, z_i) + \lambda \sum_{c=1}^k \|w_c\|_1, \\
\text{s.t.} \quad & \sum_c w_c = \vec{0}; \sum_c b_c = 0,
\end{aligned}
\tag{2.7}
$$

where λ is a regularization parameter, $\|w_c\|_1$ can be interpreted as the reciprocal of the geometric margin $m_c = \inf\{\|x - y\|_\infty : f_c(x) = 0, f_c(y) + 1 = 0\}$, defined as the L_∞-distance between two hyperplanes $f_c = 0$ and $f_c + 1 = 0$. Here m_c measures separation of class c from the remaining classes.

The L_1-penalty shrinks the estimated coefficients and coerces some small coefficients to be exactly zero. Therefore, for sufficiently large λ, or sufficiently small s, many estimated coefficients $\hat{w}_{c,j}$ become exactly zero, which enables the L1MSVM to perform feature selection within the framework of classification.

The feature selection aspect of L1MSVM is particularly useful to data with the number of features p greatly exceeding that of observations n. Although the solution of (2.5) may not be unique when p exceeds n, Theorem 2.1 shows that the solution $\hat{w}$ is unique for sufficiently small s and can be uniquely defined when s is large.

Theorem 2.1. *Let $l_s(w, b) = \sum_{i=1}^n V(f(x_i), y_i)$ and $t^* = \inf\{\sum_{c=1}^k \|w^*_c\|_1 : l(w^*, b^*) = 0\}$. Then, the solution of (2.5) is unique with probability 1 when $s < t^*$, provided that the distribution of $x \in \mathrm{R}^p$ is absolutely continuous in the Lebesgue Measure.*

For $s > t^*$, $\hat{w}(s)$ is defined as $\hat{w}(t^*)$, when $\min_{w,b} l_s(w, b)$ first reaches 0, yielding a well defined $\hat{w}(s)$ for all s. The following lemma shows that $\hat{w}(s)$ is sparse in non-zero coefficients, implying that the number of features selected by L1MSVM never exceeds $n(k-1)$.

Theorem 2.2. *The number of non-zero coefficients in $\hat{w}(s)$ is no more than $2n(k-1)$.*

2.3 Tuning and computation. Key to the performance of L1MSVM is the choice of tuning parameter s, which controls the tradeoff between training and generalization, in addition that it determines the number of features used in classification. Adaptive selection of s is necessary to maximize the generalization performance of L1MSVM. In application, selection is usually performed based on cross-validation, which is applied to different values of s to seek the optimal s yielding the best performance. In this process, (2.6) is solved repeatedly with respect to s. This is computational intensive, particularly in a high-dimensional problem.

The memory required for computation is also of concern. For any fixed s, L1MSVM solves (2.6), which is a linear program of dimension $2(p+1)k+n(k-1)$. When p is in the order of thousands, a standard package, such as "lpsolve" in R, may fail to allocate memory required for computing. Therefore, L1MSVM is not feasible to handle large problems without an efficient algorithm.

Motivated by Zhu et al. (2003) and Hastie et al. (2004), we develop an efficient algorithm that constructs the entire path of solutions $\hat{w}_{c,j}(s)$ of (2.5) for all possible values of s simultaneously. The basic idea is to locate breakpoints of $\hat{w}_{c,j}$, and to determine the corresponding right derivative of $\hat{w}_{c,j}(s)$, denoted as $d_{c,j}(s)$, at each joint. Let s_l be the l-th breakpoint. Once $\hat{w}_{c,j}(s_l)$ and $d_{c,j}(s_l)$ are determined, $\hat{w}_{c,j}(s)$ can be obtained by $\hat{w}_{c,j}(s)=\hat{w}_{c,j}(s_l)+d_{c,j}(s_l)(s-s_l)$ for all $s_l < s < s_{l+1}$, due to the property of piecewise linearity. This greatly reduces the computational cost: there is no need to solve (2.6) at each s, as long as we can compute $\hat{w}_{c,j}(s_l)$ and $d_{c,j}(s_l)$. We briefly describes the algorithm in analogy to parametric linear programming.

Algorithm:

Step 1: Start with the optimal solution at the first breakpoint $s_1=0$ where $w_{c,j}=0$.

Step 2: Increase s from lth break point s_l, find the next breakpoints s_{l+1}, where either a $\hat{w}_{c,j}(s)$ becomes zero or a $\sum_{j=0}^{p} w_{c,j}(s)x_{ij}+1$ becomes zero.

Step 3: At $s=s_{l+1}$, let either a $\hat{w}_{c,j}(s)$ become nonzero or a $\sum_{j=0}^{p} w_{c,j}(s)x_{ij}+1$ become nonzero, and compute the right derivative at s_{l+1}

Step 4: Repeat step 2 and 3, until the solution can not be improved.

Our algorithm permits rapid computation of adaptive selection of s. It effectively reduces the memory requirement and makes computation of high-dimensional problems feasible, since L1MSVM selects no more than $n(k-1)$ features, and at most nk variables are required to be stored in computing.

3 Numerical studies

3.1 Simulation. This section examines the performance of L1MSVM with respect to its generalization accuracy and feature selection in both simulated and benchmark examples. We compare it against OVA.

A four category classification problem is considered. First, $(u_{i,1},\dots,u_{i,20})$ is sampled from $N(0, I_{100\times 100})$; $i=1,\dots,80$. Second, 80 instances are randomly assigned to the four classes, with 20 instances in each class. Third, a linear transformation is performed: $x_{i,j}=u_{i,j}+a_j$; $j=1,2$ and $x_{i,j}=u_{i,j}$; $j=3,\dots,100$, with $(a_1,a_2)=(d,0),(0,d),(-d,0),(0,-d)$ for classes 1-4, respectively, where three values $d=1,2,3$ are examined. In this example, only two features x_j; $j=1,2$ are relevant to classification, whereas the remaining 98 features are redundant.

For both L1MSVM and OVA, the tuning parameter is optimized over a discrete set in $[10^{-3},10^{3}]$ with respect to the test error over an independent test sample of size 20,000, which well approximates the generalization error. Their optimal test errors, as well as the number of selected features, are averaged over 100 replications, and are summarized in Table 1. Here, the smaller number of selected features tends to indicate better selection.

Table 1: Average test errors and the standard errors (in parenthesis) as well as average number of features selected and its standard errors (in parenthesis), over 100 simulation replications for L1MSVM and OVA.

Distance		Test error	# Feature
$d=1$	OVA	56.87(0.25) %	67.17(1.93)
	L1MSVM	42.20(0.09) %	2.20(0.05)
$d=2$	OVA	16.21(0.09) %	5.72(0.38)
	L1MSVM	15.18(0.04) %	2.06(0.02)
$d=3$	OVA	3.50 (0.02) %	2.51(0.13)
	L1MSVM	3.35 (0.02) %	2.02(0.01)

With regard to prediction, L1MSVM performs much better than OVA when $d=1,2$, and slightly better when $d=3$, in view of their standard errors. This may be explained by presence/absence of the dominating class. When d is small, the four classes overlap largely. In this situation, OVA may suffer from the difficulty of lack of the dominating class. With respect to feature selection, L1MSVM outperforms OVA with respect to feature selection in that it removes more redundant features, and surprisingly, it selects nearly the true model even in a difficult situation with largely overlapping classes. In contrast, OVA selects more redundant features, which seems to agree with our discussion regarding the difficulties of OVA in the Introduction.

3.2 Benchmark example. We now examine the performance of L1MSVM on four benchmark examples: Wine Recognition, Glass Identification, Multi-feature Digit Recognition and Satellite Image, and compare it to OVA.

The first three examples are available from the UCI Machine Learning Repository at *www.ics.uci.edu/˜mlearn/MLSummary.html*, whereas the last one can be found at *www.liacc.up.pt/ML/statlog/datasets/satimage/satimage.doc.html*.

The Wine Recognition example comprises three classes of 178 instances with 13 features. The Glass Identification example has six classes of 214 instances with 9 features. Here, the two-degree polynomial decision functions $f_c(x) = \sum_{j=1}^{13} w_j x_j + \sum_{1 \le i < j \le 13} w_{ij} x_i x_j$ are used, involving 90 features, instead of linear representations. The Multi-feature Digit Recognition example consists of ten classes of 2000 instances with 76 features. Each class corresponds to a digit from $0, \ldots, 9$. In this example, we choose samples corresponding to digits 3, 5 and 8, which yields a three-class problem. The Satellite Image example comprises of seven classes of 6435 instances with 36 features. Samples are chosen from classes 1, 2, 3, and 4, which yields a four-class problem.

For the Wine Recognition and Glass Identification examples, instances are divided randomly into two parts of roughly equal size in each class, for training and testing. For the Multi-feature Digit and Satellite Image data, 50 instances are randomly selected from each class for training, with the rest for testing. In all cases, the tuning parameter is optimized with respect to the test error over a set of pre-specified values according to the best performance, and the smallest test error and the corresponding number of selected features are recorded. The test error of each method is averaged over 100 randomly sampled sets of training and testing samples for each example. The averaged smallest test errors are reported in Table 2, as well as the average number of selected features. Also displayed are the test errors in Figure 2.

With regard to prediction, L1MSVM is superior to OVA in the Wine, Digit and Image examples, but is slightly worse in the Glass example. With regard to feature selection, L1MSVM selects less features in the Wine and Digit examples, while selecting more features in the Glass and Image examples. Here, unlike the simulated example, the effect of feature selection can not be examined in absence of the truth. For instance, in the Image example, L1MSVM achieves the better performance with more features than OVA, while in the Wine and Digit examples, the better performance is realized with a smaller number of features.

Table 2: Average test errors in percentage as well as the numbers of selected features, and their standard errors (in parenthesis) over 100 replications for OVA and L1MSVM.

Data Class×Dim		Test error	# Feature
Wine	OVA	1.55(0.13)%	9.53(0.17)
3×13	L1MSVM	1.36(0.13)%	9.24(0.22)
Glass	OVA	30.71(0.35)%	30.50(0.44)
6×90	L1MSVM	31.64(0.43)%	42.86(0.87)
Digit	OVA	7.30(0.23)%	29.65(0.86)
3×76	L1MSVM	6.23(0.20)%	28.80(1.21)
Image	OVA	8.07(0.06)%	24.08(0.39)
4×36	L1MSVM	7.64(0.05)%	28.56(0.23)

4 Discussion

This article proposes a novel L1MSVM. In contrast to OVA, the proposed methodology uses a "single machine" treatment as opposed to a sequence of machines, and overcomes the difficulties of OVA as discussed earlier. An algorithm has been developed to compute an entire solution path, as well as to reduce the computational cost of the selection of tuning parameter s. This makes the L1MSVM more efficient for high-dimensional problems. In both simulation and benchmark examples, L1MSVM can effectively performs classification and feature selection simultaneously.

References

[1] Bertsimas, D. and Tsitsiklis, J. N. (1997) Introduction to Linear Optimization. Athena Scientific, Belmont, Massachusetts.

[2] Bradley, P. S. and Mangasarian, O. L. (1998) Feature selection via concave minimization and support vector machines, in J. Shaclik (ed.), *Machine Learing Proceedings of the Fifteenth International Conference*, Morgan Kaufmann, San Francisco, California, pp. 82-90.

[3] Boser, B., Guyon, I., and Vapnik, V. (1992) A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM Press, 144-152.

[4] Bredensteiner, E. J. and Bennett, K. P. (1999) Multicategory classification by support vector ma-

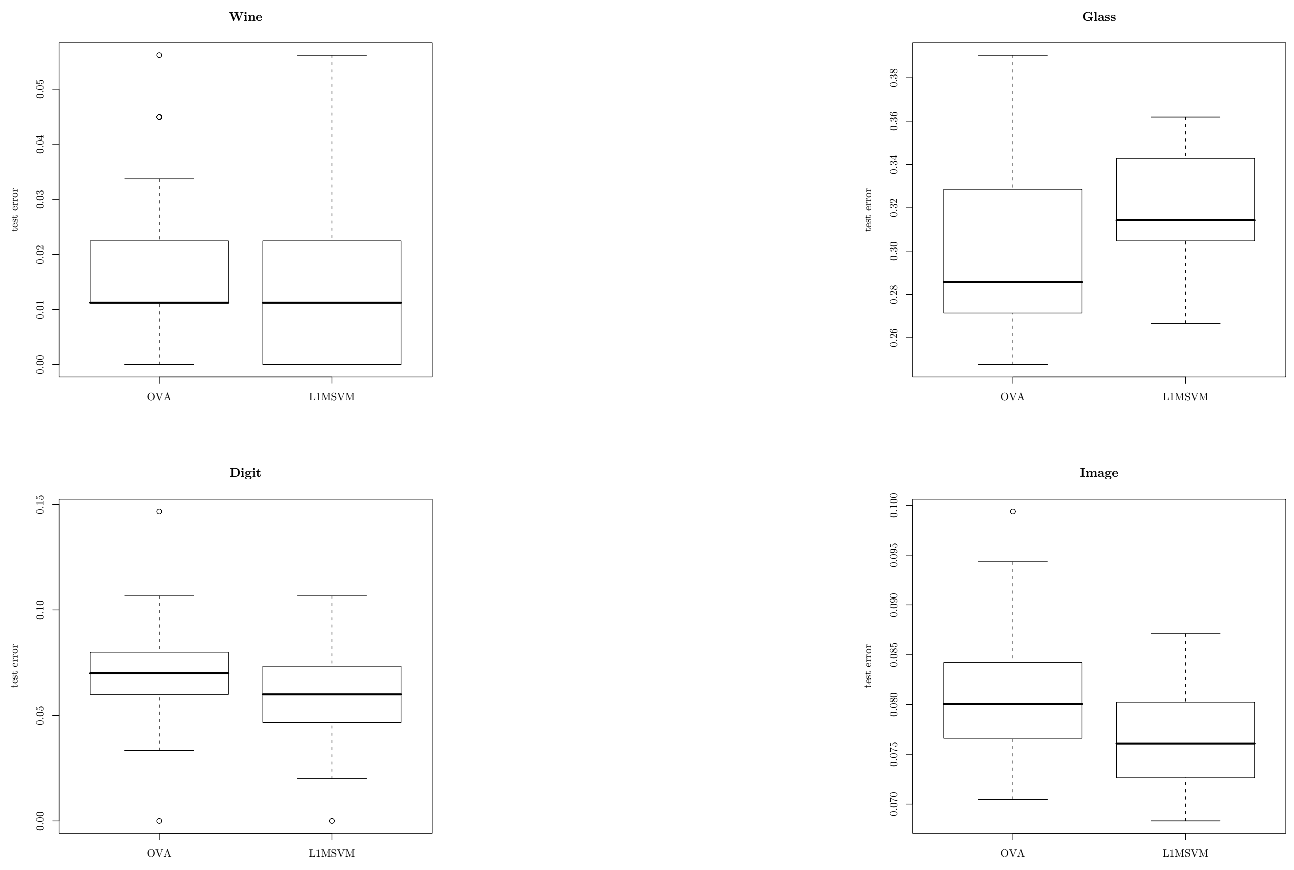

Figure 2: Boxplots of test errors for OVA and L1MSVM in four benchmark examples. Upper left: Wine; Upper right: Glass; Bottom left: Digit; Bottom right: Image.

chines, *Computational Optimizations and Applications*, **12**, 35-46.

[5] COTES, C. and VAPNIK, V. (1995) Support vector networks, *Machine Learning*, **20**, 273-279.

[6] DUDOIT, S., FRIDLYAND, J. and SPEED T. P. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, **97**, 77-87.

[7] HASTIE, T., ROSSET, S., TIBSHIRANI, R. and ZHU, J. (2004) The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, **5**, 1391-1415.

[8] GUYON, I. and ELISSEEFF, A. (2003) An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157-1182.

[9] LEE, Y., LIN, Y. and WAHBA, G. (2004) Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data. *Journal of the American Statistical Association*, **99**, 67-81.

[10] LIU, Y. and SHEN, X. (2005) Multicategory psi-learning, *Journal of the American Statistical Association*, to appear.

[11] SZEDMAK, S., SHAWE-TAYLOR, J., SAUNDERS, C.J. and HARDOON, D.R. (2004) Multiclass classification by L1 norm Support Vector Machine, In *Pattern Recognition and Machine Learning in Computer Vision Workshop*, 02-04 May 2004, Grenoble, France.

[12] TIBSHIRANI, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of Royal Statistical Society Sery B*, **58**, 267-288.

[13] VAPNIK, V. (1998) *Statistical Learning Theory*, Wiley, New York.

[14] WESTON, J. and WATKINS, C. (1999) Support vector machines for multiclass pattern recognition, *Proceedings of the Seventh European Symposium on Artificial Neural Networks.*

[15] ZHU, J., HASTIE, T., ROSSET, S. and TIBSHIRANI, R. (2003) 1-norm support vector machines. *Neural Information Processing Systems*, **16**.